EXHIBIT 71

HIGHLY CONFIDENTIAL – ATTORNEYS' EYES ONLY AND SOURCE CODE

UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

RICHARD KADREY, et al., *Individual and Representative Plaintiffs*, v. META PLATFORMS, INC., *Defendant.*	Case No. 3:23-cv-03417-VC

REBUTTAL REPORT OF DAVID R. CHOFFNES, Ph.D.

FEBRUARY 26, 2025

Table of Contents

Introduction

1. My name is David R. Choffnes. I am currently an Associate Professor in Computer Science and Executive Director of the Cybersecurity and Privacy Institute at Northeastern University in Boston, MA. I have been a professor of computer science for more than 11 years and Executive Director for nearly four years. For approximately the last five years, I have been providing consulting services to the legal industry. These consulting services primarily involve the analysis of software systems, to understand how they operate and what data they share. I have been an active researcher in the field of computer science for the past 21 years, and I am an expert in topics that include computer networking and distributed systems. I have a Ph.D. in Computer Science from Northwestern University (2010) and am a Senior Member of the Association of Computing Machinery (ACM) since 2021. My Ph.D. thesis focused on building computer systems on top of BitTorrent. My expertise includes the analysis of peer-to-peer (P2P) file-sharing technologies, such as BitTorrent, and their implications in legal contexts.

2. My curriculum vitae (CV), which contains all publications I have authored in the past 10 years, is attached hereto as Appendix A. In the past four years, I have testified as an expert witness in one confidential arbitration and have not provided deposition or trial testimony.

3. I am being compensated at a rate of $600 per hour for my work on this matter and $1,000 per hour for any deposition testimony. My compensation is not contingent or dependent on my testimony, on the content of this report, or on the outcome of this case in any way.

4. A list of the materials that I have considered in rendering my opinions for this report is listed in Appendix B. I reserve the right to offer additional opinions based on any additional discovery or additional assignments with which I am provided.

5. **Summary of Key Points**: This rebuttal will focus on three main areas: affirmative evidence of torrenting, misleading statements in Barbara Frederiksen-Cross's report, and missing evidence that could identify the extent of Meta's torrenting, including leeching and seeding. Given the evidence provided and the particularly the limited nature of the evidence in certain areas, there is a high likelihood that additional relevant evidence would illuminate activities such as leeching, the use of torrents for Libgen.rs fiction, and sharing Plaintiffs' data with other BitTorrent peers.

Affirmative Evidence of Torrenting

6. **Explicit Admissions in Frederiksen-Cross's Report.** Frederiksen-Cross's report contains several explicit admissions of torrenting activities by Meta. This admission is crucial as it directly contradicts her overall assertion that Meta did not engage in large-scale sharing of Plaintiffs' works given what we know about how BitTorrent works.

7. Specifically, Frederiksen-Cross's report explicitly confirms that Meta used BitTorrent to download and share all of Plaintiffs' works.[1] In more detail, the report admits to using BitTorrent to download and share 666 copies of Plaintiffs' works across 194 torrents. As discussed below, there is circumstantial evidence that the extent of torrenting may go beyond these 666 copies of Plaintiffs' works. In addition, Frederiksen-Cross's report further admits the use of BitTorrent in general to download and share Plaintiffs' works.[2]

Misleading Statements and Omissions in Frederiksen-Cross's Report

8. **Network Configuration and Uploading Capabilities.** Frederiksen-Cross's claim about network configurations preventing uploading is misleading. She states, "Meta's network configurations would have blocked any connections not initiated by Meta." However, nearly every home router firewall blocks unsolicited inbound connections. The fact that Meta's firewall worked this way was not unique. It instead was standard.

9. Additionally, BitTorrent is designed to work around this configuration. Note that BitTorrent facilitates data exchanges between peers, and works best when peers are sharing data with each other in a reciprocal way (i.e., uploading data to a peer from which it is downloading data). Given this context, there are two entirely common ways that BitTorrent would engage in uploading during the entire leeching and seeding process (i.e., during the download and after the download is complete).

10. First, as admitted in the Frederiksen-Cross report, the Meta BitTorrent client may initiate connections to other leechers while the download is not complete, for the purpose of fetching pieces of torrent data from those peers. While doing so, BitTorrent is designed to provide pieces of torrent data to those leechers (i.e., upload to them) to increase the chances that the peer will continue to provide pieces for downloading.

11. Second, the libtorrent client that Meta used can enable connections that circumvent the network configuration's network blocking, via a technique called "hole punching." Specifically, libtorrent implements a Peer Exchange protocol that attempts to allow incoming connections by first attempting an outbound connection.[3] As the author of libtorrent states, "the main two approaches used by libtorrent (and bittorrent clients generally) are: commonly connected peers may introduce two NATed peers to each other via the peer exchange extension. In this mode both peers try to connect to each other simultaneously, hoping that both NATs will open up pin-holes for the ports that are being attempted." The term "NAT" refers to "network address translation" and "pin-holes" refers to hole punching as described above. The NAT configuration referenced in the above quote is essentially identical (from the perspective of how unsolicited inbound network connections are blocked, but ones initiated by Meta are not) to the one used in Meta's EC2 instances used for torrenting. More

[1] Frederiksen-Cross Report ¶94, ¶110, and Table 2.

[2] Frederiksen-Cross Report ¶80 and footnote 199.

[3] https://stackoverflow.com/questions/66377090/libtorrent-nat-traversal#66476983

details about how libtorrent opens such "pin-holes" in the network firewall are detailed in the BEP 55 standard.[4]

12. In the Frederiksen-Cross report, neither of the above uploading mechanisms was tested in the EC2 networking environment where Plaintiffs' works were torrented. In other words, Frederiksen-Cross did not test whether Meta's libtorrent client uploads data to peers when this client initiates peer connections. Nor did Frederiksen-Cross test whether "hole punching" allowed new peer connections that led to Meta's libtorrent client uploading pieces of torrent data. Instead, her only experiment involved a simple "netcat" (or "nc") command issued on BitTorrent's default port, which yielded expected results but ones that are *irrelevant to the behavior of libtorrent*.

13. In contrast, the above facts strongly suggest that Meta shared torrented content (including the Plaintiffs' works) while downloading and seeding over 200 terabytes of data via BitTorrent.

14. Further, the Frederiksen-Cross report does not discuss documents where Meta employees indicated they wanted to keep torrenting activity off of Meta infrastructure so that seeders could not be traced back to Meta IP addresses.[5] Such statements indicate knowledge of how BitTorrent shares content while downloading and seeding, and, relatedly, that employees seemed to know this conduct was not appropriate.[6] In other words, Meta employees seemed to be quite aware that their use of BitTorrent would cause Plaintiffs' works to be reshared by Meta, in contrast to the direct download attempts they had previously tried.

15. **Focus on Byte Percentages Instead of Piece Percentages.** Frederiksen-Cross's report places unnecessary emphasis on Plaintiffs' works being a small portion of each torrent when counting the number of total bytes associated with the torrent file.[7] *This is irrelevant*. BitTorrent uploads and downloads data in fixed-size chunks called pieces and does not distinguish the content associated with each piece *at all*. To BitTorrent, the sizes of files associated with each torrent do not affect uploading at all. The only thing that matters is which pieces are available to download and which ones are needed. As such, the emphasis on bytes and file sizes is misleading. Further, the percentage of pieces containing Plaintiffs' works is higher compared to the byte count, which Frederiksen-Cross's report downplays.

16. **Speculative Statements about Distribution Prevention.** Frederiksen-Cross makes speculative claims about preventing any distribution of Plaintiffs' works. She states in paragraph 100, "In sum, Meta took steps to prevent seeding data downloaded via BitTorrent, and these steps should have prevented any distribution of Plaintiffs' works by Meta." This

---

[4] https://www.bittorrent.org/beps/bep_0055.html

[5] Meta_Kadrey_00108327, Meta_Kadrey_00235448.

[6] Meta_Kadrey_00232764, Meta_Kadrey_00235868, Meta_Kadrey_00235739.

[7] She states, "The small proportion of Plaintiffs' works to the sizes of the at-issue datasets, as well as the small proportion of Plaintiffs' work pieces within the torrents they exist in, are important factors that further drive down the possibility that these works would have been seeded to other peers on the network." Frederiksen-Cross Report ¶121.

statement is speculative and not supported by concrete evidence. She ignores that uploading occurs during leeching (downloading) and seeding phases; she ignores her own statement that reciprocal uploading can happen when Meta's BitTorrent client establishes a connection to a leecher regardless of the network configuration that prevents unsolicited inbound connections; and she does not provide any evidence that the Meta's use of BitTorrent actually prevented any sharing of the Plaintiffs' works.

17. **Ignoring BitTorrent's Functionality with Multiple Peers.** Frederiksen-Cross's analysis ignores how BitTorrent works with multiple peers downloading and uploading simultaneously. In paragraphs 64 and 65, she refers to the downloading phase without any reference to the fact that all BitTorrent peers (including Meta's) are designed to upload torrent pieces to peers that are connected, interested in pieces, and unchoked. Recall that an unchoked peer is one that is selected to receive data (pieces of the torrent) already downloaded by BitTorrent. In libtorrent, the BitTorrent client used by Meta, up to 8 peers can be unchoked (i.e., uploaded to) at a time by default. Further, the report overlooks the fact that BitTorrent's efficiency and speed comes from its ability to download/upload pieces of a torrent to/from multiple peers simultaneously. As I describe below, these uploading behaviors, when considered across multiple connected peers simultaneously, leads to a high likelihood that Meta shared at least one (if not more) of Plaintiffs' works.

18. **Opportunities for Meta to Upload to Peers During Leeching Phase.** The Frederiksen-Cross report mentions that BitTorrent downloads the rarest pieces of torrent data first, but it does not recognize that the rarest pieces are necessarily the most attractive pieces of data for any other leechers in the swarm. As a result, when Meta downloaded rare pieces, they were the most likely to be uploaded by Meta to other leechers during the download phase.

19. **Omission of leeching duration.** Frederiksen-Cross's report claims that Meta's safeguards rendered the possibility of seeding highly unlikely, which is misleading. She states, "Meta's torrent download scripts limited the amount of time a torrent could theoretically have been seeding." This refers to the fact that Meta's use of BitTorrent limited *seeding* to at most sixty seconds. However, this does not account for the *leeching* phase, where BitTorrent both downloads from and uploads to other peers, and is the dominant factor in Meta's sharing of Plaintiffs' works. The torrents that Meta downloaded consisted of billions of bytes of data, likely taking multiple hours of leeching before seeding began. In fact, evidence in Meta's developer notes indicates that torrenting may have been going on for *weeks*.[8] Meta did not furnish data about when each download started and ended, thus hiding exactly how much opportunity there was for sharing data during the leeching period. As pointed out in the Frederiksen-Cross report, BitTorrent frequently changes the set of peers it uploads to and downloads from ("choking" and "unchoking" every 15 seconds), and thus during the leeching phase there were likely thousands if not tens of thousands of opportunities to share Plaintiffs' works with other peers.

[8] Meta_Kadrey_00088043, where Xiaolan Wang's update on April 1, 2024 mentioned "Anna's Archive" in "Plan for this week", April 8 update mentioned "Anna's archive Anna's Archive Logbook: - Kick off data downloading (still ongoing)," April 15 update mentioned "AA downloading continued" and April 22 reports most but not all data is downloaded.

20. **Omission of statistics to determine likelihood of piece sharing.** In the Frederiksen-Cross report, she states that it is "exceedingly unlikely" that Meta shared Plaintiffs' works but provides no statistical analysis whatsoever to support that claim. In contrast, I conducted a statistical analysis on the data considered in the report. Table 1 provides a summary of my analysis, with the first column representing the datasets mentioned in Table 4 of Frederiksen-Cross's report, and the second column representing a summary of my analysis of BitTorrent behavior under the circumstances surrounding Meta's torrenting activity. As shown below, my statistical analysis demonstrates that there is a greater than 99.99999% chance that Meta uploaded at least one piece of Plaintiffs' works to another peer. In contrast to Frederiksen-Cross's report, this analysis shows that it was exceedingly likely that Meta shared Plaintiffs' works, not exceedingly unlikely as she claims.

21. Before diving into the analysis, I will establish some facts and assumptions. First, I assume that the data in Table 4 of the Frederiksen-Cross report is correct. To ease the presentation, I copied the relevant data from the Frederiksen-Cross report into Table 1 of this report, in columns 3 and 4. Second, I assume that each torrent takes one hour to download. Given that Meta was known to be torrenting for weeks, this seems like a conservative estimate.[9] Table 2 contains the hours of downloading (or leeching) in column 2. Third, it is a fact that libtorrent uses 8 unchoke slots (i.e., unchoking is the process of uploading content to a remote peer) and that unchoke decisions are revaluated by the libtorrent client every 15 seconds. The number of times that Meta's libtorrent client evaluated unchoking decisions (and thus had an opportunity to offer a new piece to a new peer) is represented in Table 2, column 3. Last, I assume that for each peer connected to Meta's libtorrent client, there is a 50% chance[10] that the Meta libtorrent client has a piece of Plaintiffs' works already downloaded and available to share, and a 50% chance that the peer connected to Meta's libtorrent client has a piece that Meta wants.[11] In this case, the peer is unchoked due to mutual interest. For each peer at a given interval, there is thus a 25% chance of mutual interest in one piece (50% x 50%). To determine the chance of mutual interest in a piece of a torrent containing Plaintiffs' works, we multiply this 25% by the probability that a piece of a torrent contains Plaintiffs' works (Table 1, column 4). I use this to compute the chance of an unchoked peer uploading a piece containing Plaintiffs' works for each collection of torrents, and list them in column 5 of Table 2.

22. To simplify the equation below, I use "i" to represent the number of unchoke intervals, "s" to represent the number of unchoke slots, and "p" to represent the percent chance of uploading Plaintiffs' works to one peer during slot.

---

[9] See previous footnote.

[10] Why 50%? When the download starts, the Meta BitTorrent client has 0% of the pieces; when the download ends (immediately before seeding), Meta has 100% of the pieces. I'm picking the average of these extremes.

[11] The reasoning for 50% is the same as above, but for the remote peer connecting to Meta's libtorrent client.

23. Note that I use the Frederiksen-Cross report analysis directly and have not attempted to validate or reproduce those numbers. I reserve the possibility of revisiting these figures if independent analysis or additional facts change the nature or accuracy of such claims in the report.

Table 1: Overall stats for Torrents in Frederiksen-Cross report. First, third, and fourth columns are reproduced from the Frederiksen-Cross report.

Dataset	Percent Chance of Sharing Plaintiffs' Works at Least Once	# Torrents Containing Plaintiffs' Works	Percent of Pieces Containing Plaintiffs' Works
Libgen.rs Non-Fiction	72.91%	2	0.136%
Internet Archive	99.999574%	46	0.056%
Z-Library	99.99999%	146	0.023%

Table 2: Values used in statistical analysis.

Dataset	Hours Leeching	Unchoke Intervals (i)	Unchoke slots (s)	Percent chance of uploading Plaintiffs' works to one peer during slot (p)
Libgen.rs Non-Fiction	2	480	8	0.034%
Internet Archive	46	11,040	8	0.014%
Z-Library	146	35,040	8	0.00575%

24. To calculate the probability that Meta shared at least one piece of Plaintiffs' works, I focus on a Bernoulli experiment,[12] where the probability of BitTorrent picking a piece of Plaintiffs' works is fixed and statistically independent. To simplify the math, I consider the simple case of finding the probability that Meta's BitTorrent client did not share a piece of Plaintiffs' works in any of the upload slots considered during the leeching phase. This probability, P(n), is defined by:

$$P(n) = q^n$$

where q is the probability of not uploading a piece of the Plaintiffs' works during an unchoke slot and n is the number of slots considered during the leeching phase. The value of n is equal

[12] https://en.wikipedia.org/wiki/Bernoulli_trial

to the number of unchoke intervals (i) times the number of unchoke slots (s); the value of q is (1- p), where p is the percent chance of uploading Plaintiffs' works to one peer during slot.

25. The probability that Meta *did* share at least one piece of the Plaintiffs' works (which I call "U") is

$$U = 1 - P(n) = 1 - (1-p)^{(i*s)}$$

26. The value of the equation ranges from 0 to 1, corresponding to a range of a 0% chance to a 100% chance.

27. According to the first row of Table 2, the probability that Meta shared at least one piece of the Plaintiffs' works in Libgen.rs Non-Fiction is:

$$1 - (1- 0.00034)^{(480*8)} = 0.7291$$

In other words, there was a 72.91% chance that Meta shared a piece of Plaintiffs' works in Libgin.rs Non-fiction.

28. According to the second row of Table 2, the probability that Meta shared at least one piece of Plaintiffs' works in Internet Archive torrents is:

$$1 - (1- 0.0000575)^{(11040*8)} = 0.99999574$$

In other words, there was a 99.999574% chance that Meta shared a piece of Plaintiffs' works in Internet Archive torrents.

29. According to the third row of Table 2, the probability that Meta shared at least one piece of the Plaintiffs' works in Internet Archive torrents is:

$$1 - (1- 0.00014)^{(35040*8)} = 0.9999999$$

In other words, there was a 99.99999% chance that Meta shared a piece of Plaintiffs' works in Internet Archive torrents.

30. While Frederiksen-Cross claimed that such uploading of Plaintiffs' works was "exceedingly unlikely," this simple statistical analysis suggests that it was, in fact, far more likely than not. In the case of Plaintiffs' works contained in Z-Library and Internet Archive, it was nearly a foregone conclusion.

Evidence Not Provided by Meta

31. **S3 Storage Data**. Frederiksen-Cross's report does not address whether S3 storage data was reviewed, which could provide additional evidence of torrenting activities. Further, the report did not explain how and why Meta produced the list of torrented data files for the report, nor what may have been omitted from the search via this process.

32. **Evidence of likely Libgen torrenting**. The Krein report states, "Meta's script for torrenting from Anna's Archive includes an 'Example command,' which specifies a 'dataset_name' of 'libgen_rs_fic,' as well as an 'input_path' […] and an 'output_dir'", the mentioned directories are where "LibGen fiction (i.e., 'libgen_rs_fic') is the data torrented." In the Frederiksen-Cross report, she states, "In evaluating the text of comments, it is important to understand that they are discretionary annotations that a programmer can record in a script or program. They are not functional code, and they do not control any aspect of the script's actual processing."

33. However, the comment in the download_spark.py file clearly indicates torrent data associated with the Libgen.rs fiction datasets, and how they get moved to Amazon S3 storage. No justification is provided that explains why this specific, working command was in the comments, but the corresponding torrented data was not produced or discussed during interviews. Generally speaking, programmers do not provide such specific working commands as examples unless they have tested them first, i.e., they have run the command in the comment. Frederiksen-Cross's report does not speak to why this comment is there in the first place, nor whether that command to download Libgen fiction was ever executed to download that content. The evidence from interviews, command line history (see below), and S3 storage would help address this issue, but is currently missing.

34. **Command History Files**. Linux systems like the ones used to download and share Plaintiffs' works by default store a history of the commands issued by the users of those systems. Given that the torrent files were specified by command and not in the code of download_spark.py, the command history files could shed light on the specific commands executed for torrenting. These history files were not provided.

35. **Data transfer details.** The Frederiksen-Cross report references the EC2 network security configuration that blocked unsolicited inbound connections (but that did not prevent uploading in general) but does not provide evidence from billing data that data transfers to the Internet did not happen. Such evidence necessarily exists because Amazon charges its customers based on how much data is transferred out of an EC2 instance.[13] Meta should furnish this data both for periods when torrenting was not happening, and for when it was, allowing us to establish a baseline of how much data was sent from EC2 to Internet destinations before using BitTorrent, and thus establishing by comparison and with high likelihood how much data was shared with BitTorrent peers during the periods when torrenting was happening.

36. Specifically, Amazon Web Services provides their customers with a dashboard under "Billing and Cost Management" > "Cost explorer" > "New cost and usage report" where filters can be arranged to display the number of outbound GB transferred per day out of each EC2 instance. I have run this on my own EC2 account and confirmed its availability for current and historical data.

[13] https://docs.aws.amazon.com/cur/latest/userguide/cur-data-transfers-charges.html

37. **File creation/modification time data.** Information about when torrented data files were created, and when they were last modified, can identify the period when downloading was occurring. This information has not been provided.

38. **Multiple versions of code.** Frederiksen-Cross refers to multiple versions of code in paragraph 77 of her report, but only one copy of one version of code has been furnished for inspection.

39. **Libgen download details.** According to Meta_Kadrey_00107954, the following text indicates that torrenting was used for 10TB of Libgen data in 2024: "Libgen (10 TB out of 10 TB): we got almost all we want (all torrents posted after 2023-03-01) with a few ones pending." Frederiksen-Cross, however, does not mention this data at all. There needs to be corresponding details provided to explain this discrepancy.

40. **Interviews with developers**. The Frederiksen-Cross report relies on interviews from Bashlykov and Wang, among others, but the transcripts of those interviews are not provided.

Conclusion

Summary of Key Points: This rebuttal has highlighted the affirmative evidence of torrenting, the misleading statements in Frederiksen-Cross's report, and the missing evidence of the extent of Meta's leeching and seeding.

Respectfully Submitted,

David R. Choffnes, Ph.D.

February 26, 2025